

CS112 –  
Java  
Programming

Fall 2022

Copyright 2022 Paul Haskell. All rights reserved.

# Interesting Java Topics

## Notes

1 week delay for Lab06, till Monday Sept 26. We will go over it this week

- THANK YOU to everyone who put in a lot of work on it already!

Today's homework will be to work on FixCapitalization.java and Plagiarism.java again

- We will go over these today in class
- Due date is Fri Sept 23 11:59pm
- Keep in Lab05 directory!

Quiz03 this Wednesday

Midterm Weds Oct 5th

## Class Review

Java syntax

Computer Science concepts

How to design and write programs

Three main categories of material in CS 112: syntax, concepts, writing programs

- A little more Java syntax every day
- Concepts e.g. "references and objects", encapsulation
- How to design programs is mostly YOUR PRACTICE. We will do examples in class.

## Syntax review

Basic variable types: double chanceOfRain = 0.85;

Type conversions: automatic or with cast: float rain = (float) chanceOfRain;

Classes:

```
class MyClass {  
    // Class variables  
    long classMember;  
  
    // Constructors  
    MyClass(float x) { classMember = (long) x; }  
  
    // Methods  
    long getValue() { return classMember; }  
}
```

## Syntax Review

### Object creation and use:

```
MyClass anObject = new MyClass();  
System.out.println( anObject.getValue() );
```

**Boolean expressions:** `&&`, `||`, `!`, `==`, `!=`, `>=`, `<=`, `>`, `<`

```
boolean done = (counter >= maxValue);  
boolean programCanExit = done && !foundErrors;
```

`while()` loops, `if()` statements

Boolean: true or false

USEFUL ONLINE TUTORIAL:

<https://testautomationu.applitools.com/java-programming-course/>

## Syntax Review

### Constants:

-17

- How to make long? Short? Byte?

110.34

- How to make float?

What are these?

- 'v'
- "w"

What are Boolean constants?

V: char. W: string. Booleans: true, false

USEFUL ONLINE TUTORIAL:

<https://testautomationu.applitools.com/java-programming-course/>

## Last time...

We discussed Objects vs References

And in the lab we worked with arrays, like

- `String[] args;`
- `Card[] allMyCards = new Card[52];`
- `allMyCards[0], ..., allMyCards[51]`

Arrays are like in Python  
Up to element #51, not #52

## Today – Wrapper classes







A few more Java built-in classes

## "Wrapper" classes

Byte, Short, Integer, Long, Float, Double, Character, Boolean

- **Class** types that "put a wrapper around" the built-in types

### Why use these?

- Sometimes you need a class type e.g. to be stored in Java's advanced storage data structures (arrays, trees, dictionaries, etc)
- Or we want "advanced" functions in the class types, e.g. `Character.isLetter()`, `Character.isDigit()`, `Character.toLowerCase()`

## Special capabilities of Wrapper Classes

### AUTOBOXING

```
Integer myValue = 1; // how do we mix Integer and int?  
void doubleMe(Integer x) { x = x * 2; }
```

Similar to the String operation:

```
String myStr = "Look, no call to new!";
```

AUTOBOXING: automatic casting, but only for the Boxed classes. CANNOT DO THIS WITH YOUR OWN CLASSES! (e.g. Doubler)

But more trickiness!

```
void doubleMe(Integer x) { x = x * 2; }
```

```
Integer myValue = 1;
```

```
doubleMe(myValue);
```

```
System.out.println(myValue); // what does this print?
```

It prints 1

- Why??

The "x = x \* 2"; statement does not change the value of our Integer object

It creates a new Integer object and has x (but not myValue) point to it

But more trickiness!

Another example

```
void fixMe(String s) { s = s + " (sometimes)"; }  
String myString = "I love Java!";  
fixMe(myString);  
System.out.println(myString);
```

What does this print?

"I love Java!"

String variables never change values! New String objects are created

But recall...

```
void ProcessNumbers(Doubler ddd) {  
    ddd.set(20.0);  
}  
  
Doubler myDoubler = new Doubler(); myDoubler.set(1.0);  
ProcessNumbers(myDoubler);  
System.out.println(myDoubler.get()); // prints 40.0
```

Why?

We programmed `class Doubler` so its stored value can change.

Java's built in classes (String, Byte, Integer, ...) are immutable.

You just have to remember this.

Immutable. Cannot change value of an object. CAN create a new object

## Quiz02

I will return quizzes after picking up from TA's

Let's go thru some of the problems



Lab07



## Homework status

First—don't forget—this course is curved.

I am reviewing homeworks with CS dept chair...

...but students are having problems

Let's spend more time learning how to tackle the homework

Lab06 delayed till Monday Sept 26

I am replacing planned Lab07 with **FixCapitalization** and **Plagiarism** from Lab05 (again)

- Second chance to work on it
- We will review problems together now
- Due Friday Sept 23
- Keep in Lab05 directory!

If you got perfect score on Lab05, nothing required!